CSCI 620 – Intro to Big Data

**Assignment #8 – Clustering**

**Name:** *Yuvraj Singh*

## 1) K-Means Normalization

Throughout the assignment, the movie documents that were manipulated were documents whose type was movie, their number of votes greater than 10,000, and has a startYear and avgRating field. The first section of the assignment was to perform a K-Means Normalization for those movies with the first position being the normalized start year and the second position being the average rating. In the script, *assignment8.py*, the method *def kmeans_norm* grabs all the movie documents that meet the above criteria and creates a new field in those documents called kmeansNorm with their normalized values.

## 2) Creating Centroids

For the second section of the assignment, a method had to be created that will select k random documents from genre g and using their kmeansNorm value to create a new document in a collection called Centroids. These documents will be assigned ids starting from 1 to k. The method *def centroids* takes in two arguments, k and g, where k is the number of centroids and g is the genre of the documents to compute.

## 3) Clustering

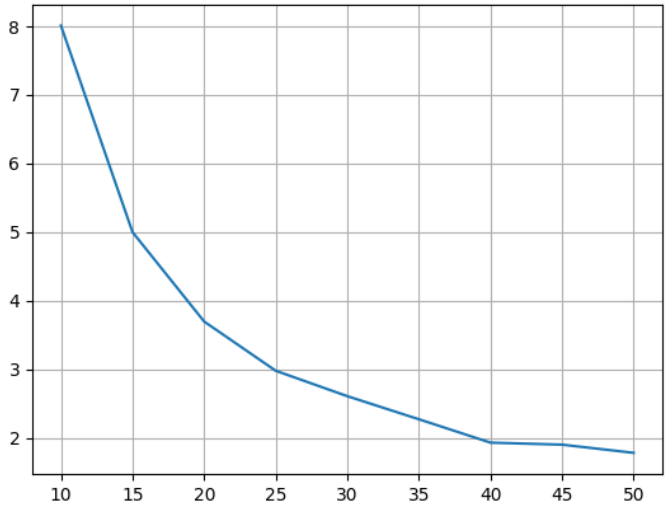
The next section of the assignment was to implement one step of the k-means algorithm which is to assign a new field called ‘cluster’ to each document which represents the _id of the closest centroid. Afterwards, the documents in the centroid collection is updated with the new centroids.

## 4) Running K-Means for each Genre

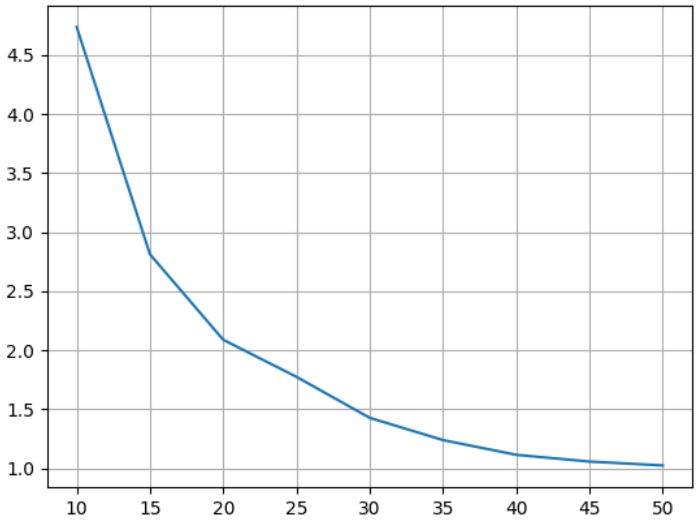
For this section of the assignment, a set of genres were given, Action, Horror, Romance, Sci-Fi, and Thriller. For each genre, a set of centroids were to be created and then the k-means had to be performed

for 100 iterations. This was to be repeated for a new set of centroids, starting from 10 centroids up to 50, with a step of 5. For each genre, the sum of squared errors had to be plotted against the value of k. The method *def clustering* will loop for each genre and perform the clustering and calculate the sum of squared errors. At the end, a plot will be generated for each genre. For each plot, the X represents K whereas the Y represents the SSE.

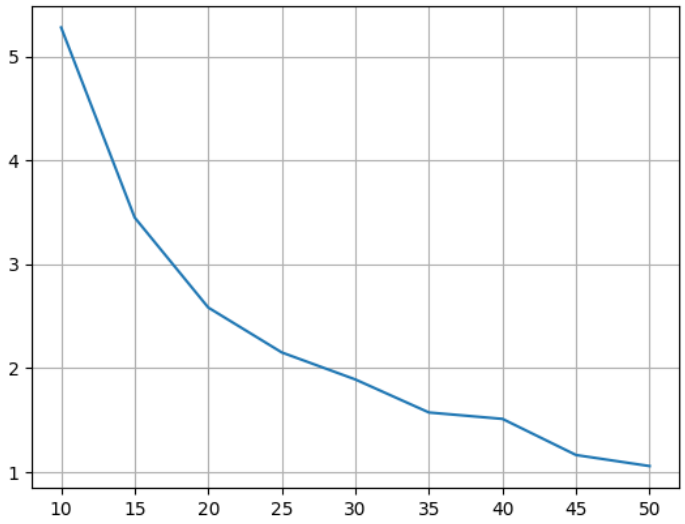
**Action**



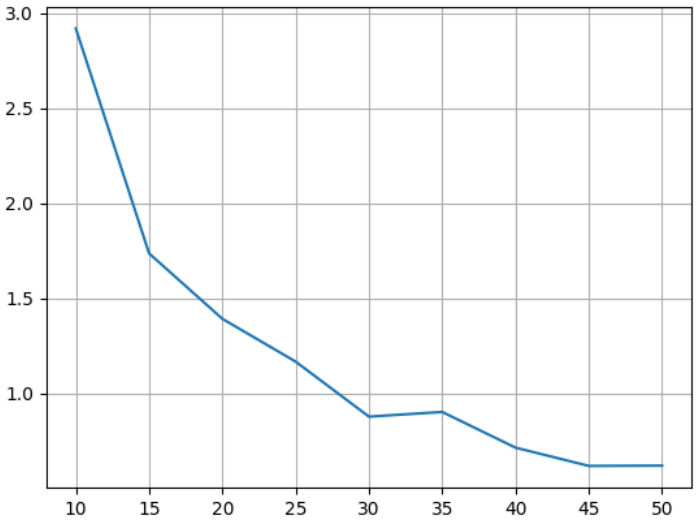

**Horror**



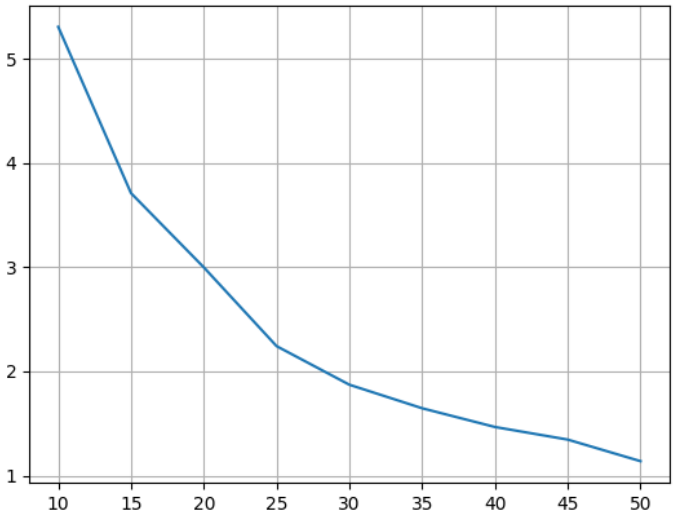
## Romance




## Sci-Fi



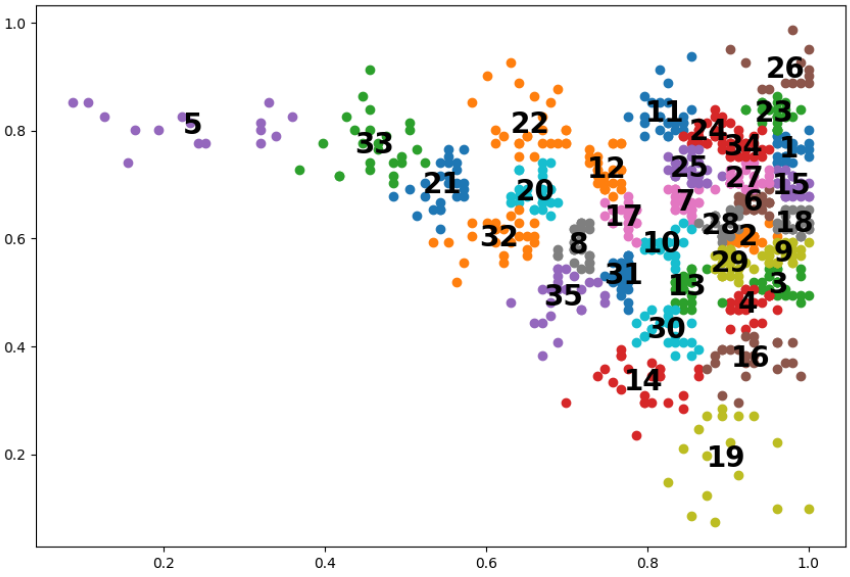
# Thriller



### 5) Selecting Best Cluster

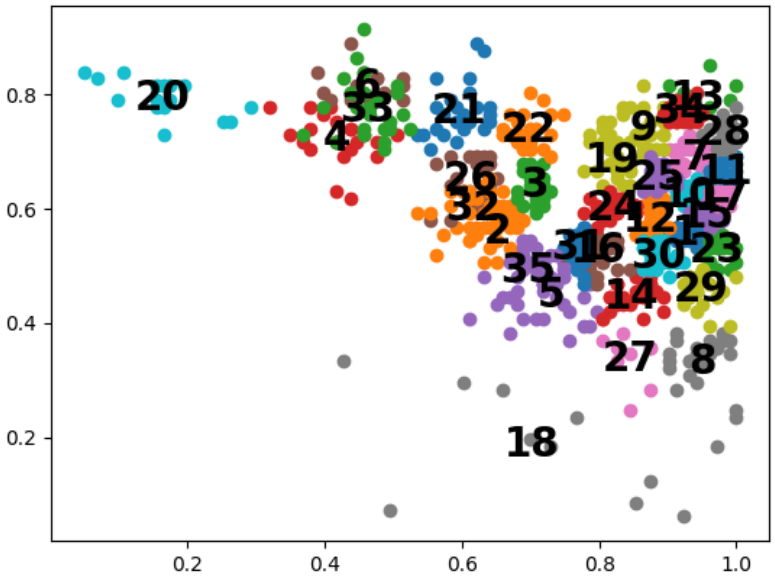
For each genre, based on the plot generated in the previous section, a K had to be selected which would best represent the clusters for the genre. For each genre, only at most 20 samples were plotted for each centroid because using more will crowd the graph and will make it difficult to interpret. The method *def best_clusters* performs the clustering for a specific K for each genre and will then plot the clusters and the centroids. For each genre, a K was hard coded into the method. Such K was chosen based on the previous graphs in Section 4. For each plot, the X represents the normalized start year and Y represents the normalized rating. The plot below represents the 'Action' genre with using a K of 35. With this we can see the centroids evenly spread out.

Action



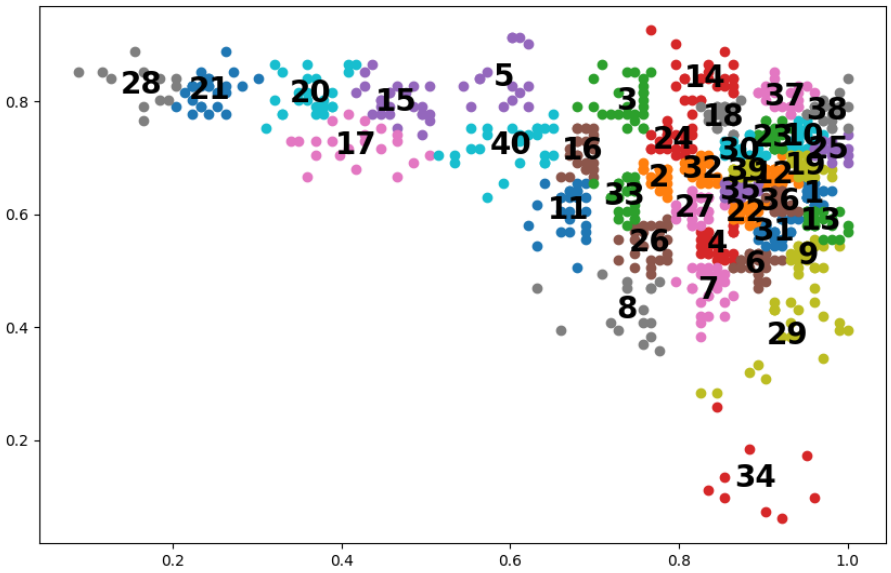
The next plot represents the ‘Horror’ genre with using a K of 30. Even though a smaller number of centroids were chosen compared to the ‘Action’ genre, the clusters here are more tightly packed and closer to each other.

Horror



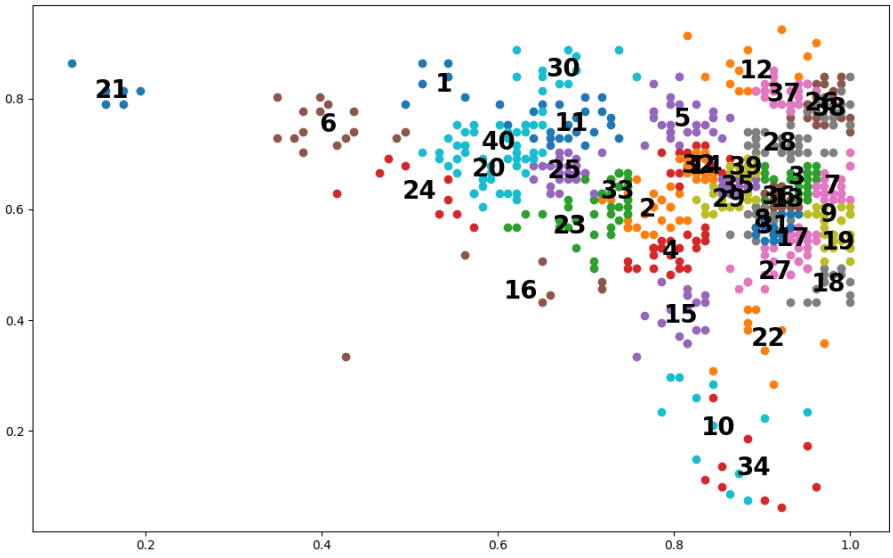

For the ‘Romance’ genre with K=40, the centroids are a little bit more spread out compared to ‘Horror’.

Romance



With the ‘Sci-Fi’ genre with K=30, the clusters are still a little bit closer to each other in the middle of the right side, but here we can see a cluster completely on the other side of the others by itself.

Sci-Fi



The ‘Thriller’ genre was the genre with the highest number of centroids with K=50. With such a high number of centroids, we can see that each cluster is neatly packed together with not much skew expect for a few of them

Thriller